Chung Shan Medical Journal 2023; 34:67-74

airiti

Original Article

A Gesture Recognition System Based on a Dual-stream Model for Deep Learning

Hao-Chuan Chu[1], Yong-Xian Wang[2], Ting-Yu Wang[3], Tzu-Hao Chang[4], Cheng-Hung Lin[5], Yen-Ching Chang[6*]

[1] Department of Medical Informatics, Chung Shan Medical University, Taichung City, Taiwan
[2] Department of Medical Informatics, Chung Shan Medical University, Taichung City, Taiwan
[3] Department of Information Center, Chung Shan Medical University Hospital, Taichung City, Taiwan
[4] Department of Information Center, Chung Shan Medical University Hospital, Taichung City, Taiwan
[5] Department of Information Center, Chung Shan Medical University Hospital, Taichung City, Taiwan
[6] Department of Medical Informatics, Chung Shan Medical University, Taichung City, Taiwan

Purpose: With the advent of COVID-19, dynamic hand movement recognition is getting important in the hospital. In order to reduce the risk of contact infection, we develop a gesture recognition system for the electronic nursing whiteboard.
Methods: For better efficiency and accuracy, the system adopts MediaPipe technology to capture the hand features of the image and collects the 21 key points of the hand; hence, we establish the dataset of their corresponding 21 key points of the original images. For gesture recognition, we propose a dual-stream model for deep learning: one stream is constructed from one 3DCNN (3D Convolutional Neural Networks) model followed by three ConvLSTM (Convolutional Long short-term memory) models, simply abbreviated as 3DCNN-ConvLSTM, and the other is constructed from three LSTM models, simply abbreviated as LSTM. For convenience, the dual-stream model is simply abbreviated as 3DCNN-ConvLSTM+LSTM. For training, the reconstructed hand images from the dataset of all 21 key points are input into the first-stream model 3DCNN-ConvLSTM; the original 21 key points of each corresponding hand image are input into the second-stream model LSTM, and the different features of the two models are merged through multi-layer fusion technology to train the dual-stream model.
Results: For the original 3DCNN-ConvLSTM model, the accuracy performed on the grayscale images is 57.5%; for the 3DCNN-ConvLSTM model, an improved version of the original 3DCNN-ConvLSTM model, the accuracy performed on the reconstructed hand images is promoted to 95%. Furthermore, the dual-stream model 3DCNN-ConvLSTM+LSTM has a higher accuracy of 97.5% and obtains a quicker and smoother convergence. In addition, the dual-stream model has raised the transmission rate by 146 times plus.
Conclusions: Our proposed model not only raises the accuracy from 57.5% up to 97.5%, but it also needs a tiny fraction of the original transmission time, lower than 0.685%.

Keywords: Deep Learning, Gesture Recognition, Dual-Stream Model, 3DCNN-ConvLSTM, LSTM, 3DCNN-ConvLSTM+LSTM.

* Corresponding Author: Yen-Ching Chang
Address: No. 110, Sec. 1, Jianguo N. Rd., Taichung City, 40201, Taiwan
Tel: +886-4-2473-0022 ext 12212
E-mail: nicholas@csmu.edu.tw

1. Introduction

Recently, deep learning for gesture recognition

airiti

has advanced, and some related models, such as 3DCNN (3D Convolutional Neural Networks) [1–2] and LSTM (Long short-term memory) [3], have also been proposed. However, they are generally in the research stage; the trained model is seldom implemented for practical use, especially in the hospital. Before the COVID-19 pandemic, the hospital implements disinfection regularly; during the COVID-19 pandemic, all places possibly touched by hands have to be disinfected often.

As the hospital is gradually becoming more intelligent, many touch devices, such as electronic whiteboards, have been consecutively introduced. Under the use of different physicians and nurses, the risk of cross-infection is getting higher and higher. However, it isn't easy to perform timely disinfection on such electronic devices. In addition, some large-scale electronic whiteboards are too high for many doctors and nurses to touch. Therefore, many new devices were introduced, but seldom and even not used; these problems indirectly hinder the development of smart hospitals.

As we know, all operations must be performed under disinfection. When surgeons and nurses need to use computers and other devices, the risk of the operation is raised and its inconvenience is produced. In order to solve these problems, we develop a gesture recognition system based on a dual-stream model for deep learning. In the system, we use only an ordinary camera lens and a gesture control mechanism to replace the original touch control system, and therefore, no touch is necessary. It is hoped that the system can not only reduce the risk of cross-infection in the hospital and but also make the operation smooth. Since only an ordinary camera lens is needed, the proposed system largely reduces the demanding hardware cost and provides a more user-friendly operating environment.

In order to lower the background noise of the system and promote its transmission rate, we adopt MediaPipe technology [4] to capture the features of the hand image and collect the 21 key points of the hand. Through the 21 key points of the hand, we can reconstruct its corresponding hand skeleton image.

In order to detect and classify dynamic hand gestures from multi-modal data, Molchanov et al. [5] proposed an online detection and classification system, called a recurrent 3DCNN.

Zhang et al. [6] first proposed the original 3DCNN-ConvLSTM model, a combination of 3DCNN and ConvLSTM, which can learn 2D spatiotemporal feature maps. The learnt 2D feature maps can further be encoded into the global temporal features and local spatial features. In the paper, we will adopt an appropriately improved version of the original 3DCNN-ConvLSTM model [6] to learn spatiotemporal features for capturing gesture recognition.

For gesture recognition, Al-Hammadi et al. [7] proposed a 3DCNN for sign language recognition. The input of the 3DCNN model is a series of RGB images captured from an ordinary camera. For better performance, in the paper, we will use the reconstructed hand images instead of RGB images as our three-dimensional training set.

In order to capture the spatial and temporal dependences, Ercolano and Rossi [8] proposed a kind of CNN-LSTM, CNN followed by LSTM, for daily life recognition. The CNN-LSTM model can learn the spatial dependencies between the limbs in a 3D-grid skeleton representation and also learn the temporal dependencies between continuous patterns.

Although these papers mentioned above can be effectively used to identify human activities, there is still a significant gap between simulation and application. In order to practically recognize human activities, especially hand activities, we will apply the MediaPipe technology [9] to the frontend for better accuracy and a higher transmission rate. For better performance, we will add another stream model, constructed from three LSTM models, simply abbreviated as LSTM, to strengthen the recognition effects. Through multi-layer fusion technology [10] for integrating the advantages of two streams (3DCNN-ConvLSTM and LSTM), a dual-stream model, abbreviated as 3DCNN-ConvLSTM+LSTM, will be proposed in the paper.

The paper is organized as follows. Section 2 introduces the methodology, including the flowchart of the system as well as the frontend and the backend components. Experimental results and discussion are presented in Section 3. Finally, Section 4 concludes the paper with some facts.

2. Methodology

airiti

In this section, we will introduce our proposed dual-stream model for deep learning. This model features two streams or series models; the first-stream is one 3DCNN model connected in series with three ConvLSTM models, called the 3DCNN-ConvLSTM model, or simply abbreviated as 3DCNN-ConvLSTM and the other comes from three LSTM models in series, called the LSTM model, or simply abbreviated as LSTM. These two streams are then connected in parallel. The reconstructed images of the hand from the dataset of all 21 key points are input into the 3DCNN-ConvLSTM model; its original 21 key points are input into the LSTM model; and then the features of the two models are fused to train the dual-stream model, the 3DCNN-ConvLSTM+LSTM model.

2.1. The Flowchart

Our proposed architecture consists of two parts: the frontend and the backend. The frontend is for data capture; the backend is a gesture recognition system. In the frontend, we use HTML5 and JavaScript to establish a web page; we also use the MediaPipe package to capture real-time hand images and extracts the 21 key points of the corresponding hand images; and finally we transmit them to the backend. In the backend, our proposed dual-stream model is used to predict actions, and sends back the prediction results to the frontend in order to further develop required functions. The flowchart is shown in Fig. 1.

2.2. The Frontend

In the frontend, we use an ordinary camera together with MediaPipe technology to capture 21 key points of the hand for gesture recognition. This architecture allows us to distinguish static images without transmitting the images to the backend for judgment, thereby largely reducing the burden of transmission. Based on the principle of the architecture, many basic control functions have also been developed successively.

In the recognition process of dynamic hand images, a large number of images generally need to be simultaneously transmitted from the frontend to the backend for prediction. The image files transmitted to the backend must be further decoded before they can be used, but this process requires high computational costs. For efficiency, the image files transmitted to the backend were replaced by their corresponding hand skeleton image files, each image connected by 21 key points of the hand. Therefore, the frontend only needs to transmit the data of all the corresponding 21 key points instead of the data of all the images to the backend. For reconstructing each image, we use its corresponding 21 key points to draw the skeleton picture of the hand, thereby lowering the expensive transmission cost and saving the time of decoding image files.

2.3. The Backend models

Our proposed system for gesture recognition features two streams or series models combined in parallel. One series model is one 3DCNN model followed by three ConvLSTM models, simply abbreviated as 3DCNN-ConvLSTM, and the other is three LSTM models in series, simply abbreviated as LSTM. They serve different purposes and are introduced in the following subsections.

2.3.1. 3DCNN-ConvLSTM

The 3DCNN-ConvLSTM model is composed of one 3DCNN model and three ConvLSTM models. The model integrates the advantages of 3DCNN for

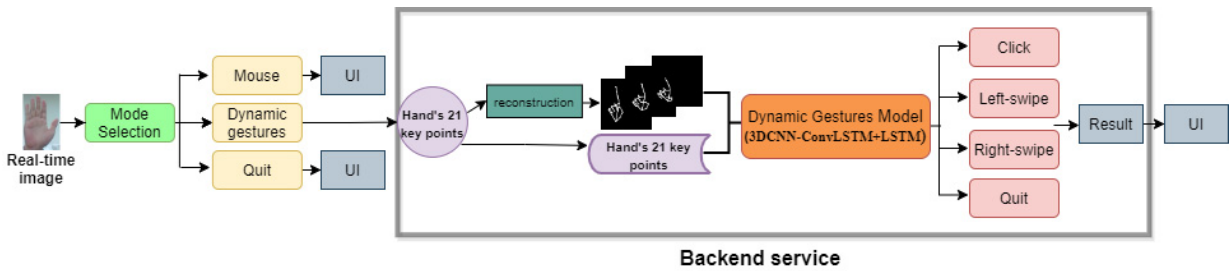


Fig. 1. Flowchart of our proposed architecture: the frontend for data capture and transmission as well as the backend for our gesture recognition system.

airiti

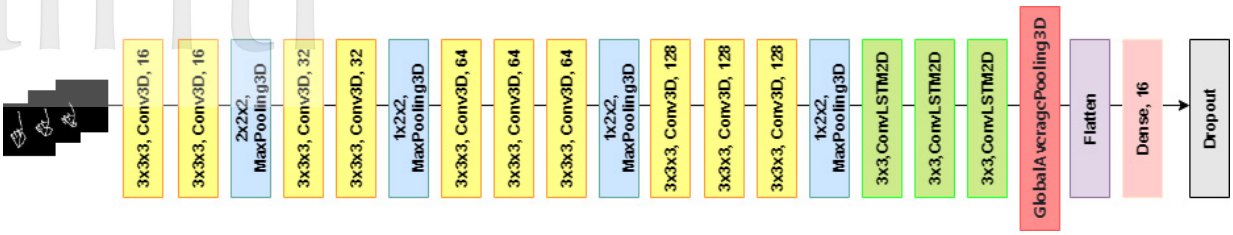


Fig. 2. Architecture of 3DCNN-ConvLSTM: one 3DCNN model consisting of 10 conv3D models and three Maxpooling3D models, followed by three ConvLSTM models.

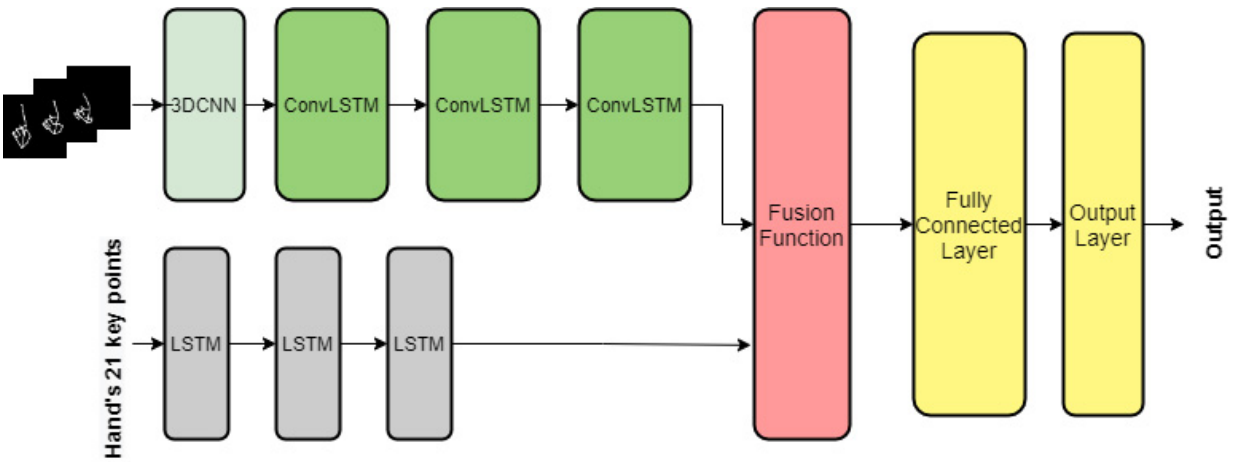


Fig. 3. Architecture of our proposed dual-stream model: the first stream is the 3DCNN-convLSTM model (also see Fig. 2); the second stream is the LSTM model; and then two streams are paralleled together.

well recognizing images and the ability of LSTM for capturing time-series features. Based on the idea of 3DCNN-ConvLSTM, we establish a suitable model as our gesture recognition system. In the system, the size of each input image is 64 × 64; a set of complete actions are composed of 25 pictures; and finally we train continuous dynamic images. The model architecture is shown in Fig. 2.

2.3.2. LSTM

The LSTM model is three LSTM models in series, and it is used to train the 21 key points of the hand. Since it is a time series problem to train the 21 key points of the hand, each set of values enters the model in a stable and continuous state and hence it is not necessary to transmit the original image of the hand. In this work, we use a succession of 25 sets of 21 key points input into the LSTM model as a complete action, and then carry out the training.

2.3.3. Our Proposed Dual-Stream Model

Since the original 3DCNN-ConvLSTM model [6] needs much time to train, we, for efficiency, propose a dual-stream model in the paper, an improved 3DCNN-ConvLSTM model and the LSTM model in parallel.

Based on the respective advantages of two models, we use multi-layer fusion technology to integrate the features of two models in the fully-connected layer to train the complete model. The architecture of our proposed model is shown in Fig. 3.

2.4. Modularization

When finishing the training of our proposed dual-stream model, we implemented modularization by the flask suite in Python so that more diversified modules and applications can be proposed in the future.

3. Experimental settings, results, and discussion

In this section, we first describe the development environment, and then introduce how to collect datasets for training and test. Finally, we discuss

airiti



Fig. 4. Dataset: four types of actions; a set of 25 continuous images as a complete action; totally 550 sets in different angles.

the efficiency of the frontend for data transmission as well as the accuracy of the backend for gesture recognition.

3.1. Development environment

This work was performed in the same computing environment. The software was TensorFlow and python3; the hardware was a CPU processor (Intel Core i7-9750H) with 24 GB RAM and a GPU processor (Nvidia GTX1650).

3.2. Datasets

We set up a simulated field in the Department e set up a simulated field in the Department of Information Center, Chung Shan Medical University Hospital, and then shot all the simulated actions. All the shoots were focused on the hand. A complete action consists of 25 continuous images. Meanwhile, we used MediaPipe to draw a skeleton picture of the hand and collected the corresponding 21 key points of the hand. Therefore, we have two types of datasets—a dataset of RGB images of the hand and their corresponding 21 key points.

For better performance, the collected data will be pre-processed. The original image size of 300 × 300 was reduced to a size of 64 × 64 for efficiency. The RGB images were transformed into grayscale images for lowering background noise.

Finally, the dataset consists of four types of actions—click, left-swipe, right-swipe, and quit; each action has 25 continuous images and was shot from 550 different angles; and therefore a total of 55,000 images were shot. The dataset for four types of actions are shown in Fig. 4.

3.3. Discussion on efficiency improvement on the frontend

Since the aim of the frontend is for data capture and transmission, the size of data will affect the transmission rate. For the original 3DCNN-ConvLSTM model [6], we need to send to the backend 25 continuous images as a package. For our proposed dual-stream model, MediaPipe technology was considered, and hence we send to the backend only their corresponding 21 key points as a package, which is a much smaller data size. Figure 5 shows that the file size is reduced from 4.7 MB to 32 KB, a ratio of 146 plus, thereby significantly lowering the transmission time.

3.4. Discussion on accuracy improvement on the backend

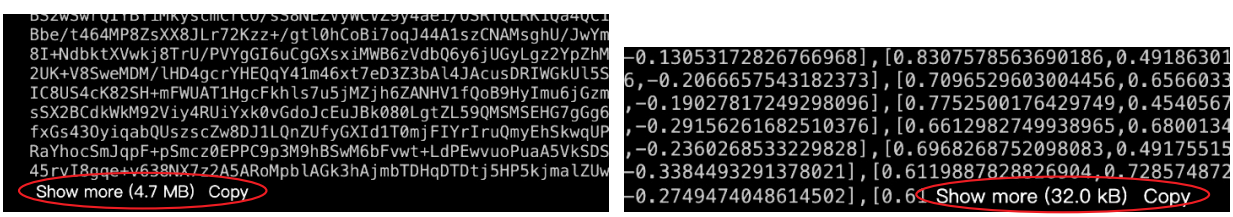


Fig. 5. Comparison of capacities required for each package transmission: (a) hand image and (b) its corresponding 21 key points.

airiti

```
Non-trainable params: 0
_________________________________________________________________
2021-08-20 14:29:52.776231: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:176] None of the MLIR Optimization Passes are enabled (registered 2)
2021-08-20 14:29:53.944787: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cudnn64_8.dll
2021-08-20 14:29:54.739241: I tensorflow/stream_executor/cuda/cuda_dnn.cc:359] Loaded cuDNN version 8201
2021-08-20 14:29:56.651986: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cublas64_11.dll
2021-08-20 14:29:57.352555: I tensorflow/stream_executor/platform/default/dso_loader.cc:53] Successfully opened dynamic library cublasLt64_11.dll
5/5 [==============================] - 7s 459ms/step - loss: 0.0428 - sparse_categorical_accuracy: 0.9750
[0.04275255650281906, 0.9750000238418579]
[[5.84695372e-04 9.95958149e-01 3.67029998e-07 3.45679140e-03]
 [5.17264241e-03 9.91291285e-01 9.14138127e-06 3.52699147e-03]
```

Fig. 6. Test accuracy rate 97.5% for the 3DCNN-ConvLSTM+LSTM model.

Initially, we used the original 3DCNN-ConvLSTM model [6] for training. The dataset was composed of 4 actions. Each action was shot 550 sets of images from different angles and each set had 25 continuous images. Each image was reduced from a size of 300 × 300 to a size of 64 × 64, and transformed from an RGB image to a grayscale image for lowering the background noise. Totally, 55,000 images were used. The experimental results show that the test accuracy of the combined model is only 57.5%.

Since grayscale images still cannot effectively reduce the background noise in the image, instead we used MediaPipe technology to capture 21 key points of the hand in the image. Based on the 21 key points of each image, we drawn its corresponding skeleton picture on the black-white image of the same size and then saved the black-white image as one of our new dataset. The 3DCNN-ConvLSTM model performed on the reconstructed dataset, the experimental results show that the test accuracy of the combined model is up to 95%.

In order to take advantage of the original 21 key points, we propose a dual-stream model for deep learning, called 3DCNN-ConvLSTM+LSTM, which combines in parallel two streams or integrated models—the 3DCNN-ConvLSTM model and the LSTM model. The first model is performed the same way as mentioned above; its input data are the skeleton images reconstructed from the 21 key points. The input data of the second model are from the 21 key points. Two models can capture different features. Through multi-layer fusion technology, two kinds of features are merged together. The experimental results show that the test accuracy of our proposed model, the 3DCNN-ConvLSTM+LSTM model, is up to 97.5%, as shown in Fig. 6.

In addition, our proposed dual-stream model needs only fewer training epochs to achieve the same accuracy as that of the 3DCNN-ConvLSTM model, thereby greatly reducing the training time. Figure 7 shows the learning convergence curves during training and validation.

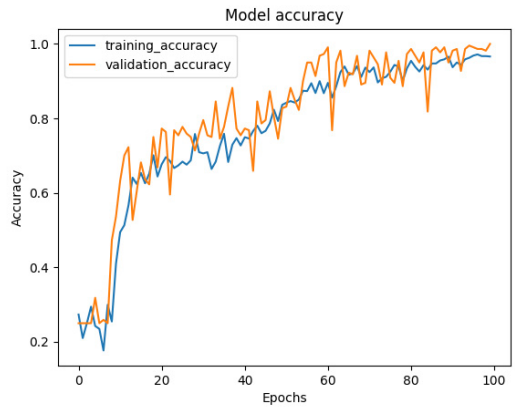


Fig. 7. Learning convergence curves during training and validation for our proposed 3DCNN-ConvLSTM+LSTM model.

4. Conclusions

In the paper, we constructed a dual-stream model, abbreviated as 3DCNN-ConvLSTM+LSTM, which features two stream models: the first-stream model is a series model of one 3DCNN model and three ConvLSTM models, abbreviated as the 3DCNN-ConvLSTM model, and the second-stream model is also a series model of three LSTM models, abbreviated as the LSTM model; the two series models were integrated in parallel together; finally, the proposed model is called 3DCNN-ConvLSTM+LSTM.

Initially, for improving transmission efficiency and lowering background noise, we introduced MediaPipe technology to capture the 21 key points of the hand in the original images, and used reconstructed black-white skeleton images to train the single-stream model 3DCNN-ConvLSTM. Experimental results show that the accuracy of the 3DCNN-ConvLSTM model is 95%, much higher than 57.5%, which was obtained

airiti

by performing the original 3DCNN-ConvLSTM model on the grayscale images transformed from the original RGB images.

In the end, for better performance, the original 21 key points were input to train and hence we introduced another model for sequential data, called the LSTM model. Therefore, a dual-stream model was established. The 3DCNN-ConvLSTM model is for reconstructed images of the original 21 key points, and the LSTM model is for the corresponding 21 key points. Experimental results show that the dual-stream model 3DCNN-ConvLSTM+LSTM is not only produces a higher accuracy of 97.5%, and it also gives a quicker and smoother convergence than the single stream model 3DCNN-ConvLSTM. In addition, the dual-stream model has raised the transmission rate by 146 times plus.

In the development process, we divided the frontend and the backend, as well as modularized various functions, thereby improving the overall flexibility of the system and making it easier to apply to future development.

At present, the work used 21 key points of the hand in the frontend to calculate the relative position of the hand and developed the virtual mouse control. In the future, we will make good use of similar principles to develop more practical functions to improve the overall service quality.

Acknowledgments: This work was supported by Department of Information Center, Chung Shan Medical University Hospital.

Conflict of Interest: The authors declare no conflict of interest.

6. References

1. Graham, B. (2015). Sparse 3D convolutional neural networks. arXiv:1505.02890.
2. Köpüklü, O., Kose, N., Gunduz, A., and Rigoll, G. (2019). Resource efficient 3d convolutional neural networks. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops.
3. Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. Neural computation, 9(8), 1735-1780.
4. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. L., and Grundmann, M. (2020). MediaPipe hands: On-device real-time hand tracking. arXiv:2006.10214.
5. Molchanov, P., Yang, X., Gupta, S., Kim, K., Tyree, S., and Kautz, J. (2016). Online detection and classification of dynamic hand gestures with recurrent 3D convolutional neural network. In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4207–4215.
6. Zhang, L., Zhu, G., Shen, P., Song, J., Afaq Shah, S., and Bennamoun, M. (2017). Learning spatiotemporal features using 3DCNN and convolutional LSTM for gesture recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 3120–3128.
7. Al-Hammadi, M., Muhammad, G., Abdul, W., Alsulaiman, M., Bencherif, M. A., and Mekhtiche, M. A. (2020). Hand gesture recognition for sign language using 3DCNN. IEEE Access, 8, 79491–79509.
8. Ercolano, G., and Rossi, S. (2021). Combining CNN and LSTM for activity of daily living recognition with a 3D matrix skeleton representation. Intelligent Service Robotics, 14(2), pp. 175–185.
9. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Uboweja, E., Hays, M., …, and Grundmann, M. (2019). MediaPipe: A framework for building perception pipelines. arXiv:1906.08172.
10. Simonyan, K. and Zisserman, A. (2014). Two-stream convolutional networks for action recognition in videos. arXiv:1406.2199.recognition in videos. arXiv:1406.2199.